DOCUMENT RESUME

ED 429 543 IR 019 451

AUTHOR Rehak, Daniel R.
TITLE A Database Architecture for Web-Based Distance Education.
PUB DATE 1997-11-00
NOTE 9p.; In: WebNet 97 World Conference of the WWW, Internet & Intranet Proceedings (2nd, Toronto, Canada, November 1-5, 1997); see IR 019 434.
PUB TYPE Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Interfaces; Computer Science Education; *Computer System Design; *Computer Uses in Education; Courseware; Databases; *Distance Education; Higher Education; Individualized Instruction; Learner Controlled Instruction; Models; *Online Systems; Student Evaluation; Training; *World Wide Web
IDENTIFIERS Carnegie Mellon University PA; Course Development; *Technological Infrastructure

ABSTRACT

The goal of the Carnegie Mellon Online project is to build an infrastructure for delivery of courses via the World Wide Web. The project aims to deliver educational content and to assess student competency in support of courses across the Carnegie Mellon University (Pennsylvania) curriculum and beyond, thereby providing an asynchronous, student-centered approach to education. The system centers on a formal data model, supported by commercial database technology. This paper provides a technical overview of the system and its features, including system architecture, course and content model, pedagogical elements, database, interface, and technology base. The use of Carnegie Mellon Online in a large introductory computing competency course is discussed, as well as plans to use the system for other courses. Two figures present the Carnegie Mellon Online System Structure and the course model. (AEF)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

A Database Architecture for Web-Based Distance Education

Daniel R. Rehak
Professor, Civil and Environmental Engineering
Chair, University Multimedia Courseware Committee
Carnegie Mellon University, Pittsburgh PA 15213, USA
Email: rehak@cmu.edu

PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
G.H. Marks
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

ED 429 543

Abstract

The goal of the Carnegie Mellon Online project is to build an infrastructure for delivery of courses via the World Wide Web. The project aims to deliver educational content and to assess student competency in support of courses across the Carnegie Mellon curriculum and beyond, thereby providing an asynchronous, student-centered approach to education. The system centers on a formal data model, supported by commercial database technology. This presentation provides a technical overview of the system and its features and discusses our initial use of it in our large introductory courses.

1. Introduction

Developing Carnegie Mellon Online was initiated to move our introductory computing competency course to the Web and offer it as a distance education opportunity to our incoming students. As a result, a general system for Web-based course delivery was developed and has been in use since Summer 1996.

We have built a course-independent, database-driven, student-centered, Web-based delivery mechanism for education and training. The system has been used both on campus and to support distance education, and is designed to be scalable to both large classes and courses with different structures and needs. The system aims to deliver educational content and to assess student competency in support of courses across the Carnegie Mellon curriculum and beyond, thereby providing an asynchronous, student-centered approach to education.

Based on the initial success with the project, and the opportunities to bring TEL (Technology Enhanced Learning) aids to more courses, work is continuing on the system, and additional courses are being developed and deployed.

2. Technical Description

Carnegie Mellon Online is a system for course-independent, Web-based delivery of educational, training and assessment materials. The system generates customized content (e.g., assessments, feedback) for each individual student and tracks the student through a course while enforcing course-specific rules and policies. The system model clearly separates course educational content, course structure, course policies and individual student records.

The system is designed to be scalable to handle large numbers of courses and students, and to provide sharing of educational content across multiple courses. It includes appropriate network security and authentication. On the client side, the student needs only a Web browser and an Internet Service Provider.

All course descriptive elements and student records are stored in a database. Individualized content is dynamically computed and delivered to the student via the Web. Courses are represented in a formal model as a structured collection of the elements of a course (e.g., instructional modules, exams, tutorials, assignments), where these elements of course content are shared across courses. The model of content is augmented with information on how courses are taught and their operational rules and policies (e.g., prerequisites, grading criteria). The entire system is data driven; only basic course-independent modeling concepts are represented directly in the system model. All content and course operations are declarative information in the data model, processed by the generic system engine. The overall system structure is shown in Figure 1.

2.1 System Architecture

The system is structured into six major elements:

- **Course delivery:** providing educational content and course status information to students via the Web and tracking their progress through a course via the database.
- **Course management and administration:** managing students, classes, staff, sections, etc., independent of actual courses and course material (e.g., moving a student from section x to section y). Course management functions are accessed via the Web.

IR019451

BEST COPY AVAILABLE

- **Student administration:** grading, providing information about individual students to instructors, etc. Again, via a Web interface.
- **Content authoring:** preparing models of a course, its structure and its policies, and entering them into the database. Also, via a Web interface.
- **Operations:** daily operations and maintenance procedures (e.g., backups, system monitors, course announcements).
- **Content:** storage for educational content and operational rules for individual courses or elements of courses delivered via the course delivery system.

Wherever possible, course-specific features are modeled declaratively in the database, either as elements of the course and system model or as elements of course content. Most of the code contains no course-specific features.

2.2 Course and Content Model

A key feature is the model of courses within the system. Courses are modeled as a multi-level collection of different types of course elements. The general approach is to provide an arbitrary number of levels of information. Elements can be shared at any level within one or more courses (see Figure 2).

At the top of the hierarchy are a collection of **components** and **subcomponents**. These are typically the units which are major portions of a course. There are an arbitrary number of such units used to describe any course, with an arbitrary number of elements within each layer. These elements essentially describe how more specific content elements are combined into a course. Associated with each of the top-level units is the information describing policies and course operational data such as grading criteria, access and authentication rules, prerequisites, etc.

The units at the lowest level of the component hierarchy are denoted **modules**. Below these component elements, each of the units of a course consists of a set of **content** elements (denoted **devices**). Content elements are classified by their role. Element roles include **learning**, **assessment** and **feedback**, along with special system operational and administrative content elements. Each module unit may have an arbitrary number of each different type of these content device elements. These content elements are composed of a set of **parts**, where each part represents a single point of interaction between the student and the course delivery system.

Devices are the primarily educational content elements of a course. Each type of device has an associated set of operational rules. For example, learning devices are information delivered to the student (e.g., lectures, syllabus) without need for student input; they may be individualized but generally are shared content for all students. Assessment devices involve presenting assessment information (typically individualized), getting results from the student, grading, and using the grade results to control the student's next step in the course (e.g., passing an exam provides a prerequisite for the next topic). There are specific models of the information associated with each type of device (Figure 2 shows more details of the model for assessment devices).

Each course is a collection of the different elements of course content and descriptions from the database. The course author selects and combines them into a single course. A set of staff, teaching sections, schedule and other administrative information is associated with the content, and this is combined with a list of registered students to fully instantiate a course for Web delivery.

While the information in the model appears to be hierarchical, it is not. Each level is described separately and named globally, and elements in any level are unordered collections of elements from the lower levels. Various parameters and course rules can be entered at many of the levels and the system can use the data from one level to override the values for the same attribute from a lower level when the individual elements are combined into a specific course.

In addition to the modeling components which describe a course, the elements of the system itself are represented in the same structure, e.g., there are **system** elements at each level in the model. Similarly, there are elements for operations and management at each level in the model. The actual data modeling is generalized to accommodate such elements, i.e., at the kernel level, the system itself represents courses and internal operations with the same representations and data structures.

2.3 Pedagogical Elements

A strength of the current implementation is its ability to handle assessments. The course and content models contain a number of specific features to represent and process assessments (e.g., examinations). For example, the system can generate custom examinations for individual students based on a number of different criteria and parameters, including their prior work and results. Examinations are also defined in terms of the topics and types of questions (multiple choice, matching, value, task instructions, etc.), and individual questions include details of grading policies, instructions to the grader (when

graded by an instructor), and feedback for the student. Generated examinations are delivered to the student, and the student's work is submitted and either graded automatically or sent to an instructor for personalized grading. In either case, detailed, customized feedback is generated and returned to the student. The amount and type of feedback are determined by the policy rules associated with the assessment device and the course.

By tracking student progress versus course requirements, the system can present a customized view of a course to the student, e.g., only work for which prerequisites have been completed is presented. The interface design is task-focused and student-centered, presenting only appropriate information to the student at each step. This eliminates the need for the student to search through the course and Web to find appropriate elements to make progress towards completing the course.

Courses built to date incorporate learning and teaching elements which are hypertext and are rich in assessment materials. This is not a limitation in the system design, but rather in available content. Multimedia, simulations and active tutors can be added and are under development. These enhancements can be accommodated by the existing course and content model. Each of these items can use the state information about student progress and the overall course model to provide customized views and student-specific content within the overall course structure and operational rules.

2.4 Database

The entire information model is stored in a database. Modeling and representation of course-specific information is clearly separated from course-independent information.

The major collections of information within the database are:

- **People**: information on students and teaching staff, independent of their association with a specific course.
- **Courses**: registration-type course information, such as schedule, and associated staff and students.
- **Content:** course content such as questions, answers, learning materials, tutorials, and course evaluations. Content is not associated with a specific course but can be shared across courses. Similarly, it is not associated with an individual student, but is instantiated as necessary for the students in a course.
- **Course Models:** descriptions of selected elements of content which comprise a course, along with related course policies. This includes modeling a course in terms of its components and modules, and associating specific content with the elements within the modules. The course models also include the internal system models in the same framework.
- **Course Instances:** selection of a course model and association with people (students, staff) and course (registration) information, customization of the content (e.g., exam generation) for each student.
- **Student Records**: tracking of the progress of individual students through courses (including a full record of student access to their personal content). The record information also includes system records in the same format.
- **Media:** media files, HTML, graphics, programs, etc., used by either the system or the courses.
- **HTML:** dynamically generated HTML saved for later delivery; stored in the database for indexing and security.
- **Code:** database procedural code.

2.5 Interface

The Web interface is specific to each individual course, in terms of its organization and form; designers are free to pick a look and feel which is appropriate for specific courses. There is a single link between the interface and the course delivery system, and the link structure is course independent. Thus a different course interface could be installed and the rest of the system would remain unchanged. A new look is developed by providing new image files and screen layout. How the interface is related to the delivery of course content and system operation is also encoded in the description of the course model.

While the system is designed to be student centered, we have been constrained by the commonly deployed Web technologies available to our distance students (courses are offered to our students around the world) and thus have limited the amount of state information portrayed via the interface. The types of interfaces which we desire to deploy require both Java and Javascript, and we have not found these technologies to be robust enough yet to deploy and support across all of our target platforms. A more advanced Java/Javascript interface is under development. This interface will maintain more information on the student or client side of the Web connection, and will not only provide active feedback, but will also focus the student on only the appropriate actions at any point in a course. Thus the system will be better able to dynamically lead a student through a course.

2.6 Technology Base

Carnegie Mellon Online is a custom system constructed from standard components. The components and system conform to Internet and other standards whenever possible.

All information is stored in a commercial relational database (Oracle V7). The course and content model is represented directly within the database. A commercial Web server (Oracle WebServer 2) provides the Web interface to the system and to the database. Most of the system consists of code (Oracle PL/SQL) to provide database access via the Web, to compute custom content, and to maintain the model of courses and students. Security is provided with SSL Web transactions and site authentication is implemented with Kerberos login. Client state is maintained with secure cookies.

Students need only a basic Web browser which supports frames, e.g., Netscape 2 or above. Interfaces must work at 640x480 with 256 colors, and response should be reasonable over a 14.4kbaud dialup connection. Courses are delivered on Macintosh, Windows and Unix clients.

Our production system is deployed on a Sun server. Hardware resources dictate the size of a course which can be represented in the database and the system response rate. The current production server is a dual processor, with dual I/O subsystems, 256MB of primary memory and 20GB of disk. The configuration is designed to provide subsecond Web/database response under normal peak loads of 50 users all requesting information simultaneously. We have development and test systems which operate on NT PCs, and a small course can be delivered using a small desktop or notebook PC.

3. Use

Carnegie Mellon Online was initially developed to support our introductory computing competency course (CSW). Starting with a pilot in the Summer of 1996, this course is now fully supported and Web delivered. Many of the features of the system architecture were driven by the demands of this course. Additional courses and pilots have been or are under development.

3.1 Computing Skills Workshop

Computing Skills Workshop (CSW) is an entry-level, one credit course, required of all of our students, covering how to effectively use computing throughout the curriculum. Its aims are to: (1) introduce new students to Carnegie Mellon's computing facilities; (2) ensure that all students have a baseline of conceptual knowledge and practical skills using a variety of productivity software (word processing, spreadsheets, etc.); and (3) equip students to make effective use of computing in the service of their other courses.

CSW traditionally was a large lecture and laboratory course operating in an assembly line process of feeding students through a formal schedule of topics, assignments and an end-of-semester examination. The course is taught both semesters, but most incoming students are scheduled for the Fall semester, with a typical Fall enrollment of 1300+ students. Labs and lecture sessions are led by undergraduates (a staff of about 50), and the course runs in two dedicated computer clusters (25 seats each) from 9:00AM to 8:00PM every day.

A plan was developed to make the course modular and self-paced, letting students take any of the modules in any order, at any time, and to support it with Web-based delivery of materials, submission of work and electronic grading. By utilizing the Web, the course would be freed from the time and place constraints of our campus and schedule and many of the course management functions could be automated. While the use of the Web is essential, the goals are driven by the desire to reshape the model of the course, not how technology supports it. Benefits include enabling students to complete the course early, providing more feedback and freeing staff resources to spend more time with students who need individual attention.

First-order benefits include: (1) students can concentrate on CSW in the summer between high school and college when there are fewer competing demands and distractions; (2) with CSW successfully behind them, students will have more time for academic demands and social opportunities once they arrive on campus; (3) students will arrive on campus with certified computing skills, prepared to apply those skills to academic courses. Second-order benefits of creating a general system include: (1) providing a powerful model of Web-based education which we can use to build additional materials for full courses or modules; and (2) creating a framework in which to imbed ongoing work in cognitive tutoring, just-in-time education, automated curriculum design, and educational evaluation.

3.2 CSW Online Structure

CSW Online is primarily a system for assessment, not teaching. The content model is rich with assessments but thin on learning materials. Web materials are complemented with traditional texts. On campus, students can attend lectures to learn about the different topics in the course.

The course itself currently covers several topics or modules: word processing, spreadsheets, email, basic Unix and Emacs, and networking (Internet, File Servers, Libraries). Once a student thinks she is ready, she can take a competency examination for a topic. Passing the course requires all modules be completed.

CSW Online presents the students with a choice of activities for each module. First they can read about the competency topics included in the examination on the topic. This is the information they need to know to successfully pass the examinations. There are limited on-line learning materials, but the topic information is indexed to the textbook (if copyright for the textbook materials were available, the information would be linked to on-line books).

For each topic, there are practice and graded examinations. In the current course (Fall 1997), the examinations are situated tasks for which the student is asked to take a set of resource files and a set of instructions and produce a new document. We have explored different options for the examinations, including multipart examinations that included automatically generated and graded sets of questions to test declarative knowledge.

We have also explored different policy options, including a fully self-paced course versus a directed schedule with due dates, and have used different mixes of take-home versus in-class graded worked. All of these variants have been built by simply providing different course models on top of a common set of content materials. The current course is actually four different combinations of course models. One option is the choice of PC- versus Macintosh-specific educational content. The other is self-paced versus directed course policy. All are delivered from the same database, and with the same interface. The content is individualized for students based on their course registration.

3.3 CSW Deployment

CSW was initially developed as a pilot over the summer of 1996. We built the system and delivered two modules (40%) of the course to incoming students working at distance. From an initial pilot group of 250 students, we had approximately 100 complete a portion of the course during our month long pilot.

The summer pilot was a technical success, and in Fall 1996, we delivered two modules of CSW via the Web to a class of 1400 and traditional delivery (e-mail, bboard, etc.) was used for the rest of the course. For Spring 97 we completed the creation of all content and had all modules available for Web delivery and supported 300 students.

CSW Online is now an option for all incoming students in the summer before their enrollment. In Summer 1997, the course was available as a distance education option, and over 250 incoming students worked on the course from around the world, and some had completed it within days. In addition to the incoming students, the course was offered on campus to upper classmen who had not finished it, to students in our summer precollege program and as staff training. Total enrollment was 400. In Fall 1997 we will have 1300+ students in the course, but those who worked on it during the summer will need to complete only one new email module which requires software only available on campus.

3.4 Other Courses and Plans

Over the Summer of 1997 we offered a small placement examination for our incoming calculus students. Traditionally, all of our calculus students are given a paper examination, delivered by mail. The results of the examination and a survey are used to place the students into the appropriate course in our calculus sequence.

In the pilot, the examination and survey were converted into a small Web-based course. The course included a background survey, a practice examination used to familiarize students with Web-based assessments, and the complete placement test. The course was designed to be completed in a single Web session, and the students were given immediate feedback on their performance and placement.

Our Computer Science Introductory Programming course (C++) is being developed for on-line delivery. This is a large, intense, mainstream academic course with 3 units of credit. Passing the course with a grade of B or above is a prerequisite for our upper-level CS courses.

A pilot course was offered at a distance to selected students in the incoming class during the Summer of 1997. This course includes online lecture materials, problem sets, practice programs and examinations for each course topic. It includes a large body of content, over 1000 questions for examinations and over 300 programming exercises. The complete pool of materials

is significantly larger than would be presented to any one student. We are also investigating Web-based compilation to support students who do not have the necessary compiler on their computers, and automated grading of programming assignments. Building the course involves defining the course model and loading it, along with the course content, into the database. A new interface structure is being designed, but there no system level code or data modeling changes are required.

Completing Introductory Programming requires passing a single mastery exam. Currently, about 1% of the 1000 students enrolled each year can successfully complete the examination without taking the course. By offering the course at a distance to our incoming students, we believe that 10% to 30% of them will be able to complete the mastery exam within the first two weeks of the semester. The students will benefit from completing the course early, permitting them either to take another course in its place, or to devote more time to other activities. We do not plan to reduce the staff, but rather redeploy the resources to provide more one-on-one support.

Other courses are being developed for deployment during AY 97-98, including one on Engineering Economics and one on Art History. In addition, the project team has received inquiries from faculty wanting to develop courses in Introductory Biology, Introductory Chemistry, and Statistical Reasoning. We are also continuing to add features and capabilities to the system to better support courses and their operations.

Acknowledgments

Elements of this report were extracted from prior documents describing CSW, the CSW Online Proposal, and the 96 Summer Pilot report. In particular, credit for some of the words above goes to Chris Thyberg, Director of CSW; Phil Miller, Director of Introductory Programming; and Diana Bajzek, Director of TELab.

Support for Carnegie Mellon Online has been provided from the Office of the University President. The project is under the operational control of the Division of Computing Services.

More Information

For more information, visit our Web site: http://online.web.cmu.edu/ or contact us via e-mail at: online+@cmu.edu

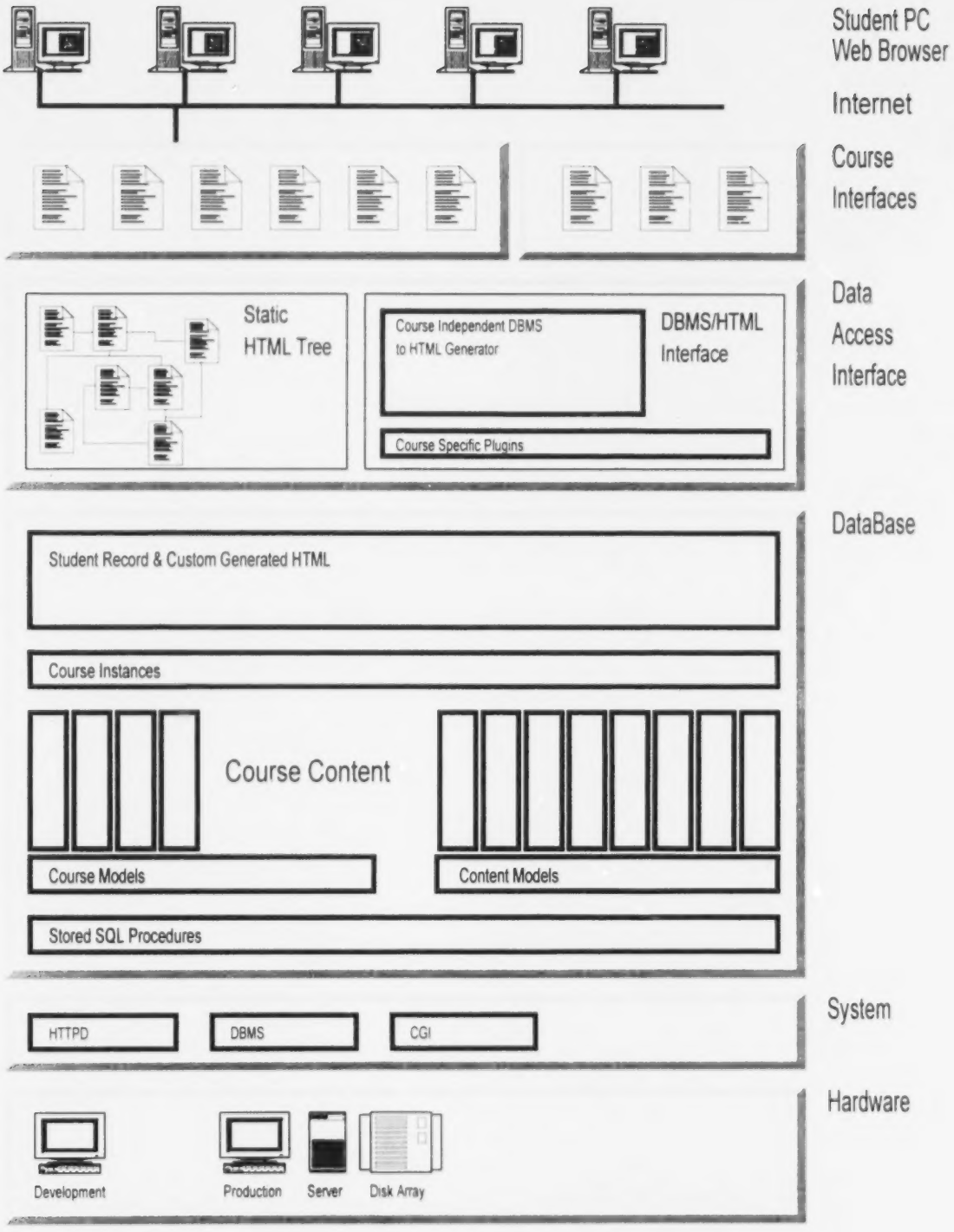

Figure 1: Carnegie Mellon Online System Structure (patent pending)

BEST COPY AVAILABLE

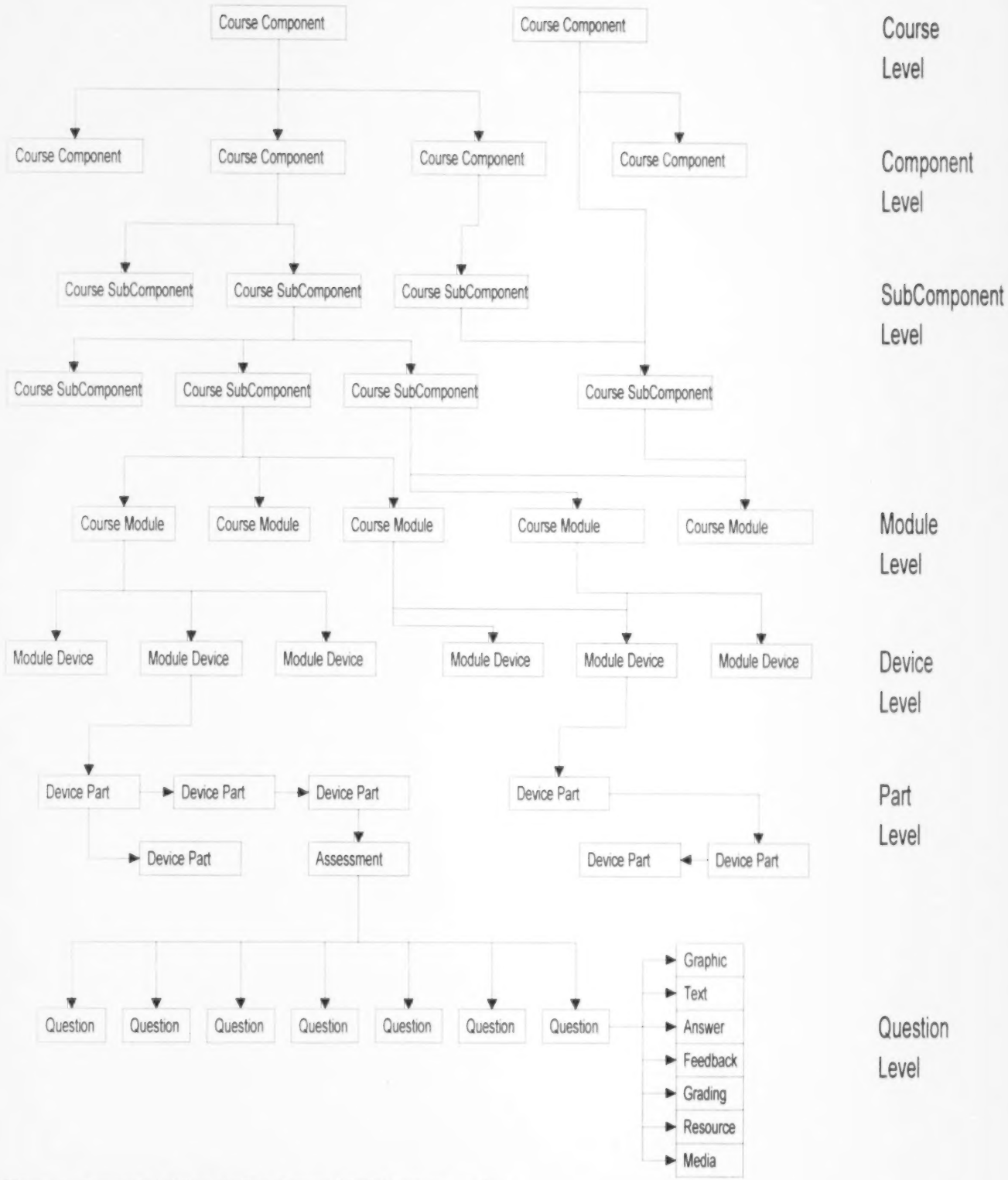

Figure 2: Carnegie Mellon Online Course Model (patent pending)